

A Nearly QP Optimized SVM for Fast CU Size Decision in HEVC Intra Prediction

Suan-Jie Cai, Yinyi Lin

Department of Communication Engineering, National Central University
Taiwan 32054, R.O.C.
yilin@ce.ncu.edu.tw

Abstract - In our previous work we employed support vector machine (SVM) to reduce computation of CU size decision in HEVC intra prediction. Experiments were conducted at medium bit rates and results revealed that negligible degradation in coding efficiency (less than 0.1% bit rate increment) can be achieved. The trained hyperplane is however not optimized for all bit rates, leading to severe loss in coding efficiency at other bit rates. In this paper instead of QP-optimized SVM we propose a nearly QP-optimized SVM for fast CU size decision that only several QP-trained hyperplanes, with friendly implementation, are proposed for fast CU size decision. The experimental results show that the proposed method can achieve nearly same coding performance as QP-optimized SVM.

Keywords: High Efficiency Video Coding (HEVC), Intra Prediction, CU Size Decision, Support Vector Machine (SVM).

1. Introduction

High Efficiency Video Coding (HEVC) uses three hierarchical unit representations (including coding unit (CU), prediction unit (PU) and transform unit (TU)) to find the best coding performance based on the quad-tree structure. In HEVC intra prediction, the CU depth level range is from 0 to 3 which correspond to the CU sizes of 64×64 , 32×32 , 16×16 and 8×8 , respectively. Instead of nine prediction modes used in H.264, the prediction unit in each CU utilizes 35 prediction modes, including DC, planar and 33 angular prediction modes. The computational load is much higher than H.264/AVC.

To reduce computation in HEVC intra prediction, N modes among 35 modes are first selected in PU based upon the rough mode decision (RMD) criterion. The selected mode number N is dependent on block sizes, with N equal to 8 for block sizes of 4×4 and 8×8 , and 3 for 16×16 , 32×32 and 64×64 . To achieve the best coding performance, the rate distortion optimization (RDO) technique is then used to search the best mode among N modes as well as the most probable modes (MPM).

To reduce computational complexities of the HEVC encoder, many papers have investigated fast intra prediction algorithms [1-3]. The results reveal that average 50% of computation time can be saved but with severe coding efficiency loss (with average 1% BD bit rate increase). The fast algorithms studied for HEVC intra prediction can be generally classified into two categories: fast CU size decision and fast intra mode decision. One technique for fast CU size decision is to employ spatially neighbouring encoded CU depth information to predict CU search arrange [1-2] to reduce computation. Parameters such as RDO cost function are also incorporated into the fast CU size decision to enhance computation saving but with more performance loss. Another technique uses both global and local edge complexities to decide the CU size [3].

2. Support Vector Machine (SVM) for Fast CU Size Decision in HEVC Intra Prediction

Unlike most papers focus on computation reduction, in our previous work [4] we proposed a fast CU size decision using SVM to maintain the coding efficiency as high as possible. In [1], the depth level of current CU is estimated using its four neighbouring CUs (left, top, top-left and top-right CUs), given as

$$D_{pre} = \sum_{i=1}^3 \alpha_i \cdot w_i \quad (1)$$

where parameters α_i and w_i represent the weighting factor and the value of depth level. α_i is equal to 0.3 for left and top CUs while 0.2 for top-left and top-right CUs. Four search arranges for current CU size decision are assigned based upon the predicted depth level D_{pre} . In [2] only top and left CUs are considered and three search ranges are assigned. Although neighbouring CU depth information provide good prediction for current CU depth search range, the experiments reveal that the search range prediction is still not adequate for some sequences and this leads to severe coding efficiency loss. TABLE I displays the accuracy of search arrange prediction using neighbouring CU size information for some test sequences. As shown, only 26% accuracy is obtained for *Vidyo1* sequence which results in 1.7% bit rate increase [4].

In our previous work we used two other parameters (variance and low-frequency AC components) in SVM to more accurately predict the search range, in addition to D_{pre} . The variance and low-frequency AC components can describe homogeneous or flat characteristics of a video sequence quite well, as shown in Fig. 1, conducted on *BasketballPass* sequence with QP=37. As shown, homogeneous or flat regions have small variance and low-frequency AC components. The variance and low-frequency AC components are respectively defined as

$$\sigma_{N \times N}^2 = \frac{1}{N^2} \sum_{j=1}^N \sum_{i=1}^N [p(i, j) - \mu]^2 \quad (2)$$

and

$$AC_{low, N \times N} = \sum_{u=1}^{N-1} |F(u, 0)| + \sum_{v=1}^{N-1} |F(0, v)| \quad (3)$$

$$F(u, v) = \frac{2}{N} C(u)C(v) \sum_{l=0}^{N-1} \sum_{k=0}^{N-1} p(l, k) \cos \frac{(2l+1)u\pi}{2N} \cos \frac{(2k+1)v\pi}{2N} \quad (4)$$

where $F(u, v)$ is the $N \times N$ DCT coefficient of a block. To maintain high coding efficiency, in our previous work only two CU depth search ranges (012) and (123) in SVM are assigned and separated using the following five features : D_{pre} , $\sigma_{32 \times 32}^2$, $\sigma_{16 \times 16}^2$, $AC_{low, 32 \times 32}$ and $AC_{low, 16 \times 16}$.

Table 1: Accuracy of CU size prediction in [1].

	Sequence	Accuracy
Class C (832x480)	RaceHorses	83.65%
Class D (416x240)	BasketballPass	74.58%
Class E (1280x720)	Vidyo1	25.83%

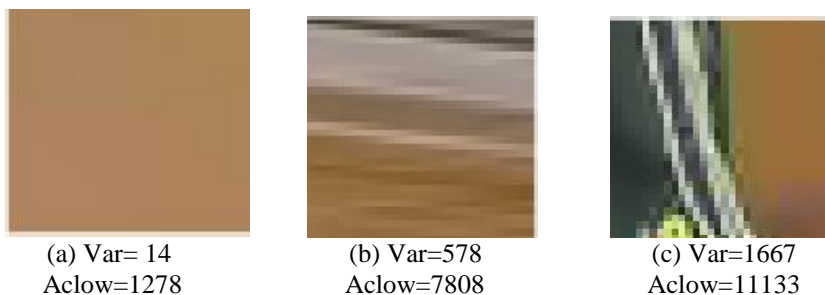


Fig. 1: Variance and low frequency AC components for homogeneous and complex blocks.

Four sequences (*Tennis*, *Basbetball Drill*, *Keiba* and *Vidyo3*) are used in SVM training for QP=30. The coding performance and computation efficiency are evaluated at medium bit rates with QP=22, 27, 32 and 37. The experimental results demonstrated that the SVM method can average reduce 16% of computation time with only 0.07% bit rate increment. It is superior to the other two methods [1-2], which have 15% and 11% timesaving, and 0.58% and 0.25% bit rate increment respectively.

Table 2: Encoded CU depth versus QP.

QP	depth 0	depth 1	depth 2	depth 3
1	0.00%	0.00%	3.54%	96.46%
5	0.00%	0.00%	0.01%	99.99%
10	0.00%	0.00%	0.87%	99.13%
15	0.00%	0.07%	3.15%	96.78%
20	0.00%	0.43%	7.74%	91.83%
25	0.00%	1.91%	14.69%	83.41%
30	0.00%	3.83%	22.82%	73.36%
35	0.57%	7.88%	34.54%	57.01%
40	1.35%	18.67%	46.18%	33.79%
43	2.30%	29.03%	48.38%	20.29%
45	4.88%	38.48%	45.27%	11.36%
47	8.37%	49.49%	37.15%	4.99%
49	12.64%	57.25%	28.60%	1.51%
50	17.15%	59.13%	22.96%	0.76%
51	18.54%	62.24%	18.83%	0.39%

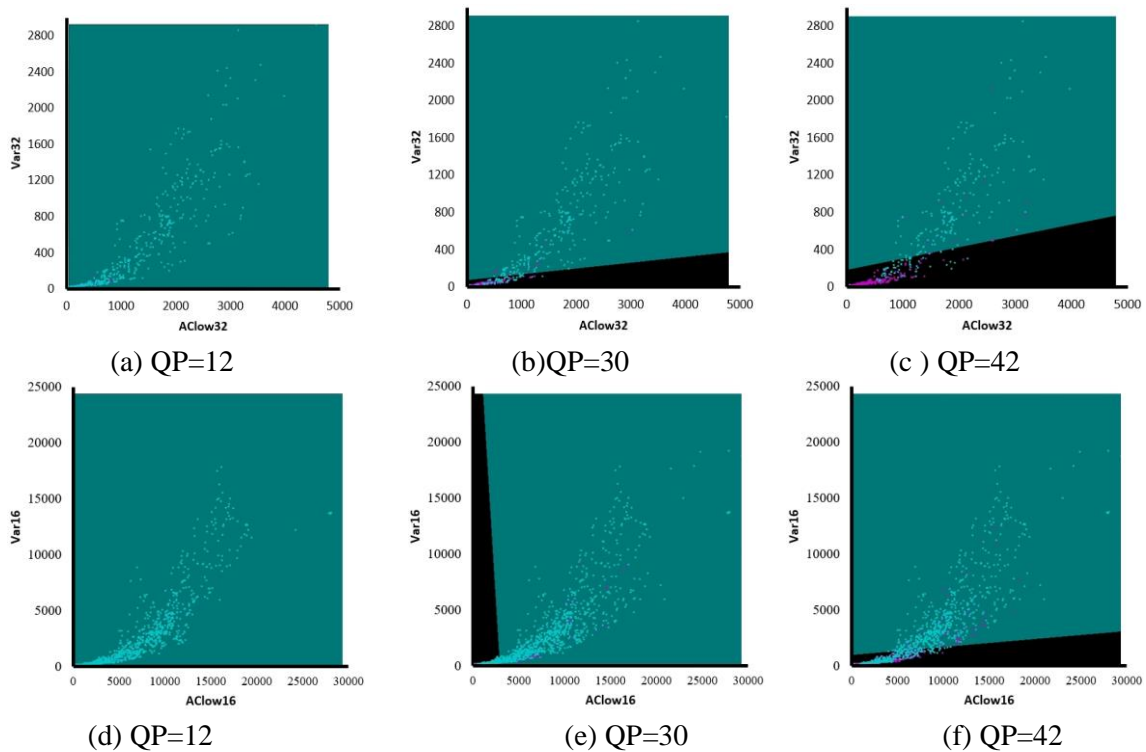


Fig. 2: Hyperplanes for various QPs.

2.1. QP Optimized SVM for CU Size Decision

Although high coding efficiency can be achieved using proposed SVM technique, the hyperplane trained with QP=30 is not optimized for all bit rates because a video sequence encoded with various QPs or bit rates possesses different characteristics and be finally encoded with different CU depth. And this leads to severe loss in coding efficiency at other bit rates. Table 2 shows the distribution of final encoded CU depth for various QPs or bit rates, conducted on *BlowingBubble* sequence. As shown, most blocks are finally encoded with small CU sizes at low QP or high bit rates while encoded with large CU sizes at high QP or low bit rates due to different characteristics. To get best coding efficiency each optimized hyperplane in SVM should be trained and used in SVM individually for each QP. For example, Fig. 2 shows optimized hyperplanes of features: $\sigma_{32 \times 32}^2$, $\sigma_{16 \times 16}^2$, $AC_{low, 32 \times 32}$ and $AC_{low, 16 \times 16}$ for some QPs (12, 30 and 42 respectively) in SVM. As can be seen, different QPs possess different optimized hyperplanes.

As QP varies from 0 to 51, the HEVC encoder requires huge memory to store all these 52 hyperplanes. In this section we investigate some QP-optimized hyperplanes for HEVC encoder instead of all 52 hyperplanes to reduce storages. The hyperplane for each QP is trained and obtained from training sequences. From experiments we observed that the hyperplanes are nearly the same for small QPs, but they are quite different for large QPs. There are 6 QP-optimized hyperplanes (QP=12, 30, 40, 45, 48, 50 respectively) are experimentally selected and used in SVM instead of all 52 hyperplanes by the least mean square error (LMSE) criteria. Table 3 displays the suitable range of QP for each QP-optimized hyperplane. For example, the hyperplane optimized using QP=12 is used as the hyperplane in SVM for QP between 0 and 20, and experimental result reveals that nearly the same coding efficiency can be achieved. To furthermore reduce computation as well as storage, we employ linear SVM method [5] for CU size decision in HEVC intra prediction in which only 63 instead of 11390 multiplication operations, as also shown in this table, is required after some manipulations. And with the simplification average 0.1% of computation time can be further saved without any loss in coding performance.

Table 3: 6 Nearly QP optimized hyperplanes used in SVM.

Range of QP	Selected QP	MUL before simplified	MUL after simplified
0-20	12	2333	8
21-35	30	2169	10
36-43	40	1686	10
44-46	45	1694	11
47-49	48	1826	11
50-51	50	1682	13
	Total MUL	11390	63

3. Experimental Results

An experiment was conducted with test model HM15.0 reference software and LibSVM3.2 [5] to evaluate both computational complexity and coding efficiency. The video sequences from classes A to E were tested and each video sequence is coded with 100 I frames. The CABAC entropy coder is employed. The linear SVM with three different hyperplanes for CU size decision is implemented into HEVC intra prediction for comparisons at low bit rates (QP=5, 10, 15, 20) and high bit rates (QP=35, 40, 45, 50) respectively. The experimental results are demonstrated in Tables 4 and 5 in which both BDBR and time saving are compared to the original HEVC intra prediction. Three cases for hyperplanes employed are investigated. The first column shows the results for the case with all 52 QP-optimized hyperplanes, and the second column for the case with hyperplane optimized with QP=30, while the third column for the proposed nearly QP-optimized hyperplanes that is shown in Table 3.

As demonstrated in Table 4, for low bit rates the SVM with proposed hyperplanes achieves nearly the same coding efficiency as the optimized hyperplanes (with average 0.03% and 0.07% bit rate increments) while the SVM using hyperplanes optimized with QP=30 has much higher bit rate loss (with average 0.31% bit rate increment), but with 14% time saving slightly better than the other two (with 12% time saving).

At high bit rates (see Table 5), the loss in coding efficiency (average 0.06% bit rate increment) using both optimized and nearly optimized hyperplanes are trifling, compared to the original HEVC intra prediction; while the SVM using hyperplanes optimized with QP=30 runs into average 0.15% bitrates increment. Average 22% of computation time can be reduced for SVM with optimized and nearly optimized hyperplanes; while only average 14% can be saved for the SVM optimized with QP=30.

Table 4: BDBR and time saving comparisons for QP=5, 10, 15 and 20.

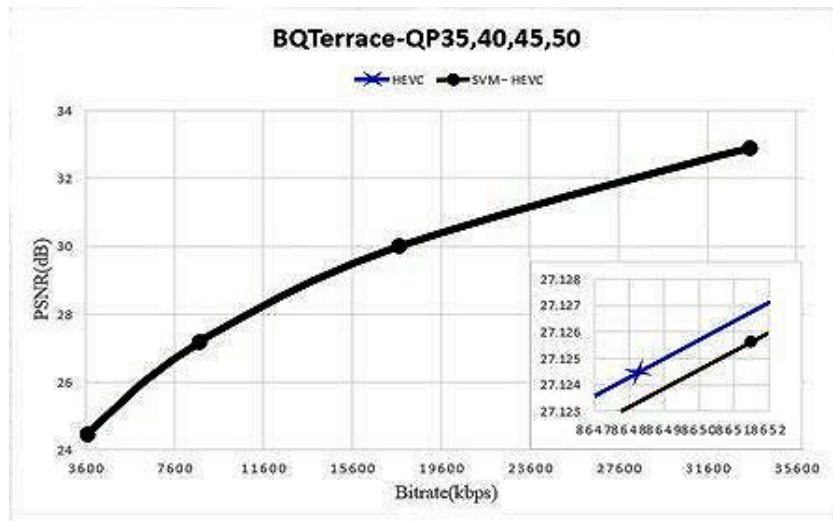
QP=5,10,15,20		BDBR(%)			Time(%)		
Sequence		Optim.QP	QP=30	Prop.	Optim.QP	QP=30	Prop.
A	PeopleOnStreet	-0.007	0.079	-0.040	-12.373	-13.938	-12.581
	Traffic	0.035	0.269	0.053	-12.387	-15.044	-12.638
B	Cactus	0.000	0.299	0.002	-11.886	-17.373	-11.094
	BQTerrace	0.108	0.355	0.103	-12.950	-17.810	-12.656
	ParkScene	0.105	0.581	0.238	-13.373	-18.324	-13.873
	KimonoI	0.052	1.448	0.487	-12.054	-15.825	-1.575
C	RaceHorses	0.000	0.045	0.003	-11.180	-12.233	-11.103
	BQMall	0.002	0.203	-0.002	-11.303	-12.774	-11.372
	PartyScene	0.000	0.002	0.000	-11.611	-10.827	-11.680
	Keiba	0.051	0.423	0.284	-14.316	-16.437	-15.581
D	BasketballPass	-0.004	0.2112	-0.004	-9.905	-12.165	-10.242
	BlowingBubbles	0.000	0.000	0.000	-9.393	-9.585	-10.071
	BQSquare	0.000	0.000	0.000	-8.651	9.292	-9.098
	FlowerVase	0.052	0.309	0.042	-9.640	-14.424	-8.817
E	SlideEditing	0.033	0.060	0.060	-13.240	-12.882	-13.335
	Vidyo1	0.064	0.496	-0.011	-14.376	-22.639	-12.426
	Vidyo4	0.025	0.462	-0.008	-14.661	-20.784	-14.469
Average		0.030	0.308	0.071	-11.959	-13.751	-11.918

Table 5: BDBR and time saving comparisons for QP=35, 40, 45 and 50.

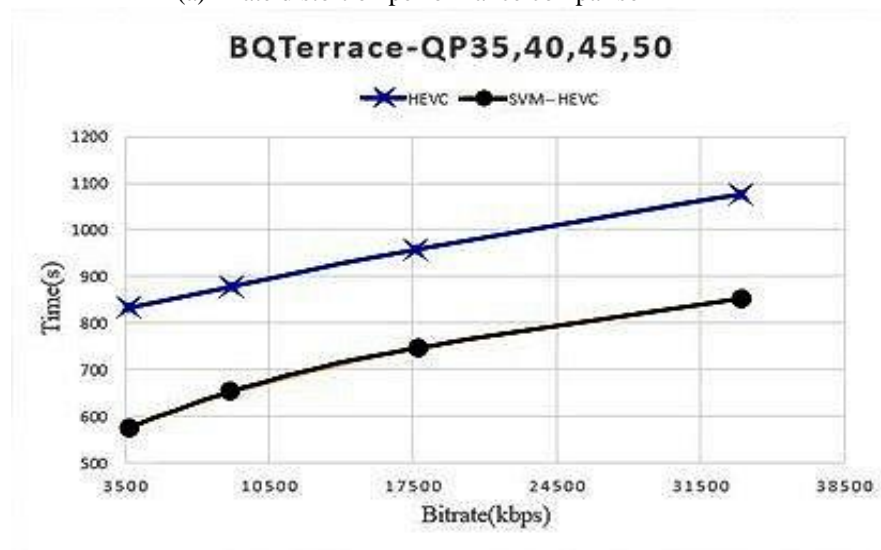
QP=35,40,45,50		BDBR(%)			Time(%)		
Sequence		Optim.QP	QP=30	Prop.	Optim.QP	QP=30	Prop.
A	PeopleOnStreet	0.003	0.009	0.004	-16.639	-14.219	-17.670
	Traffic	0.030	0.145	0.031	-24.776	-13.478	-24.716
B	Cactus	0.110	0.178	0.109	-26.328	-15.971	-26.496
	BQTerrace	0.027	0.167	0.032	-24.959	-17.270	-24.047
	ParkScene	0.096	0.165	0.091	-31.959	-17.945	-32.008
	KimonoI	0.128	0.271	0.124	-36.034	-15.823	-34.610
C	RaceHorses	0.083	0.080	0.080	-19.064	-11.852	-19.369
	BQMall	-0.010	0.022	-0.010	-18.238	-11.271	-18.962
	PartyScene	0.020	0.033	0.019	-15.910	-8.697	-16.411
	Keiba	0.115	0.162	0.118	-21.981	-16.415	-22.282
D	BasketballPass	0.008	-0.021	0.004	-22.419	-4.263	-23.004
	BlowingBubbles	-0.044	-0.055	-0.044	-11.867	-7.656	-12.820
	BQSquare	-0.008	-0.008	-0.008	-4.697	-5.759	-7.381
	FlowerVase	0.076	0.546	0.057	-27.391	-17.764	-29.636
E	SlideEditing	0.020	0.041	0.020	-13.337	-10.666	-13.249
	Vidyo1	0.147	0.325	0.141	-30.987	-23.596	-30.579
	Vidyo4	0.171	0.450	0.160	-27.714	-20.661	-27.626
Average		0.057	0.148	0.055	-22.018	-14.312	-22.404

To obtain further insight, Fig. 3 compares the rate-distortion performance and computation time for HEVC and SVM-HEVC with proposed nearly optimized hyperplanes, conducted on *BQTerrace* sequence at high bit rates (QP=35, 40, 45 and

50). As shown, SVM-HEVC can reduce about 24% of computation time with only 0.001 dB PSNR loss, compared to HEVC intra prediction.



(a) Rate distortion performance comparison



(b) Computation time comparison

Fig. 3: Rate distortion and computation time comparisons.

4. Conclusion

In this paper, we present a nearly QP-optimized SVM for fast CU size decision in HEVC intra prediction, instead of QP-optimized SVM, to reduce storage for all 52 hyperplanes. There are 6 QP-optimized hyperplanes (QP=12, 30, 40, 45, 48, 50 respectively) are selected and used in SVM instead of all 52 hyperplanes, with each QP-optimized hyperplane suits for a range of QP. The experimental results demonstrate that the proposed method can achieve nearly the same coding efficiency and time saving as the SVM with all 52 QP-optimized hyperplanes. The results also show that 12% and 22% of computation time can be saved with slightly loss in coding efficiency (with less than 0.1% bit rate increment) respectively for both low and high bit rates. We employ linear SVM method for CU size decision and after some manipulations only 63 instead of 11390 multiplication operations are necessary. The proposed linear SVM method for HEVC intra prediction is very friendly in both hardware and software implementations.

Acknowledgements

This work was supported by the Ministry of Science and Technology (MOST), Taiwan, under Grant MOST 106-2221-E-008-024-MY3.

References

- [1] L. Shen et al., "Fast CU Size Decision and Mode Decision Algorithm for HEVC Intra Coding," *IEEE Trans. Consumer Electronics*, vol. 59, no 1, pp. 207-213, Feb. 2013.
- [2] Y. Wang et al., "A Fast intra coding algorithm for HEVC," in *Proc. ICIP*, pp. 4117-4121, 2014.
- [3] B. Min et al., "A Fast CU Size Decision Algorithm for the HEVC Intra encoder," *IEEE Trans. Cir. and Sys. for Video Tech.*, vol. 25, no 5, pp. 892-896, May 2015.
- [4] H. H. Hsu, S. E. Huang and Y. Lin, "Computational Complexity Reduction for HEVC Intra Prediction with SVM," in *Proc. GCCE*, pp. 900-901, Oct. 2017.
- [5] LIBSVM - A Library for Support Vector Machines. [Online]. Available: <http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html>